

# Lesson Plan: AI as a Coding Coach

AI + Coding Starter Kit | Teacher Resource | Teachers

**Purpose:** Students learn to use AI responsibly while learning to code. The lesson emphasizes explanation, debugging, testing, revision, and student ownership. Students compare shortcut-style coding prompts with coaching-style prompts and practice asking AI for help in ways that build understanding.

**Standards summary:** This resource may support Tennessee Computer Science Foundations standards when used as part of AI literacy, computer science, programming, digital ethics, or cybersecurity instruction. Detailed standards connection appears at the end of this document.

Item	Details
Time	45-60 minutes
Subject Fit	Python, Computer Science Foundations, AI literacy, digital ethics
Essential Question	How can AI help me understand and debug code without doing the coding for me?

## Lesson Purpose

Students learn to use AI responsibly while learning to code. The lesson emphasizes explanation, debugging, testing, revision, and student ownership. Students compare shortcut-style coding prompts with coaching-style prompts and practice asking AI for help in ways that build understanding.

## Learning Objectives

- Explain the difference between AI as a coding coach and AI as a code replacement.
- Use AI prompts that ask for explanation, debugging guidance, and practice rather than finished answers.

Apply a structured debugging process to identify, fix, verify, and document a coding problem.

Reflect on whether AI use strengthened or weakened their own understanding.

- Explain code behavior and a debugging decision in their own words.

## Materials

Python editor, browser-based IDE, or teacher-provided code environment

## Python Debugging Activity handout

- Ask AI Better Coding Questions handout

## Responsible AI Use in Coding Rubric

## Projector or screen

## Optional approved AI tool demo

### Lesson Flow

#### 1. Warm-Up: When Code Breaks (5 minutes)

- Ask students: What do you usually do when code does not work? What helps you learn, and what only helps you finish?

#### 2. Mini-Lesson: Coach vs. Replacement (10 minutes)

Explain that AI can explain code, interpret errors, suggest what to check, and generate practice examples. It should not write full assignments or replace the student's reasoning.

#### 3. Prompt Comparison (10 minutes)

- Students compare shortcut prompts with coaching prompts. The class identifies which prompts build coding skill and which prompts bypass learning.

#### 4. Debugging Practice (20 minutes)

Students work through the Python Debugging Activity. They identify the intended behavior, read or run the error, ask for coaching-style help, revise the code, test the fix, and document what changed.

#### 5. Share and Discuss (5-10 minutes)

- Students explain one bug, one prompt that helped, and how they verified the fix. Emphasize explanation over speed.

#### 6. Exit Ticket (5 minutes)

- Students complete: One coding question AI can help me with is... One coding task I should still do myself is... One way I verified my fix was...

### Sample Prompt Comparison

Shortcut Prompt	Coding Coach Prompt
Do my Python assignment for me.	Explain what this error message means and point me to the line I should check first.
Write a complete number guessing game.	Help me plan the steps for a number guessing game, then let me write the code.
Fix my code and give me the final answer.	Ask me questions that help me find the bug before showing a fix.
Give me working code.	Explain what this code does line by line and give me a similar example to practice.

### Assessment

#### Completed debugging activity

## Quality of coaching-style AI prompt

Successful test or clear explanation of remaining bug

Written reflection on what AI helped with and what the student still did independently

## Exit ticket explanation

### Teacher Notes

This lesson works best when students must explain the code and the fix. A working program is not enough. Students should be able to describe the bug, the reasoning behind the change, and how they tested the result.

## Detailed Tennessee Standards Connection

This lesson directly supports standards when students use a structured debugging process, explain code behavior, test revisions, and reflect on ethical AI use in programming.

Standards source: Tennessee Department of Education, Computer Science Foundations (C10H11), May 2023. Confirm final alignment against local district pacing, approved course placement, and teacher directions.

This resource may support the following Tennessee standards when used as part of AI literacy, computer science, programming, digital ethics, or cybersecurity instruction:

- CSF 9.2 - Troubleshooting Process: Students use a structured process to identify a problem, gather information, isolate causes, test a solution, verify the result, and document what they learned.
- CSF 13.1 - Social, Legal, and Ethical Issues: Students identify responsibilities related to ethical technology use, academic integrity, copyright, appropriate AI use, and responsible programming support.
- CSF 16.1 - Programming Language: Students explore programming languages such as Python and explain how programmers use them to solve a variety of IT problems.
- CSF 16.2 - Software Development Life Cycle: Students connect planning, coding, testing, refinement, deployment, and maintenance to an iterative software-development process.

Cautious guidance: Alignment depends on local district pacing, approved course placement, teacher directions, and how the resource is used as part of instruction.